

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

5

APPLICATION PAPERS

10

OF

15

CONRADO BLASCO ALLUE,

PAUL KIMELMAN,

ANDREW BROOKFIELD SWAINE

20

AND

RICHARD ROY GRIENTHWAITE

25

FOR

30

FORCED DIAGNOSTIC ENTRY UPON POWER-UP

BACKGROUND OF THE INVENTION

Field of the Invention

5 This invention relates to the field of data processing systems. More particularly, this invention relates to diagnostic mechanisms within data processing systems which are dynamically switchable between an operational mode and a powered down mode.

Description of the Prior Art

10 It is known to provide data processing systems with diagnostic mechanisms which can be used to perform diagnostic operations (e.g. software and hardware fault identification and analysis (debug)) upon the data processing systems so as to assist in the development of hardware, operating systems, application programs, overall system designs and the like. As data processing systems become more complex and include larger numbers of elements operable in more diverse ways, the need for comprehensive diagnostic tools able to perform diagnostic operations upon the systems in a wide variety of states and configurations becomes more important.

20 Another development within data processing systems aimed at reducing power consumption, so as, for example, to improve battery life, is the use of dynamic switching between operational and powered down modes. As an example, within an integrated circuit different portions, such as different processor cores, may be selectively powered up and powered down as the demand for the processing operations they can perform varies with the user's requirements.

30 In order for diagnostic operations to be as accurate as possible it is important that the system behavior they are attempting to investigate should be perturbed as little as possible by the investigation itself. In this context, when seeking to obtain diagnostic information concerning power-up and power-down behavior it is desirable that the circuits concerned should actually power-up and power-down rather than this merely being simulated. However, a problem arises in that diagnostic circuitry

embedded within the data processing circuits being powered up and powered down will lose its configuration and so be unable to continue the diagnostic operation after the system is powered down. Continuously powering the diagnostic elements within the circuit portion being powered down is undesirable since it requires a non-contiguous power domain.

SUMMARY OF THE INVENTION

Viewed from one aspect the present invention provides apparatus for processing data, said apparatus comprising:

a data processing circuit operable to execute program instructions to perform data processing operations during an operational mode, said data processing circuit being dynamically switchable between said operational mode and a powered down mode; and

a diagnostic circuit operable to perform diagnostic operations upon said data processing circuit;

wherein when said data processing circuit returns to said operational mode from said powered down mode and any required processing operations have been completed, said diagnostic circuit prevents execution of further program instructions until released by said diagnostic circuit.

The present technique recognizes that an important area of behavior to be subject to diagnostic operations is that immediately following return to an operational mode from a powered down mode. The difference in processing speed between the circuitry under investigation and typical diagnostic circuitry means that the typical diagnostic circuitry is not sufficient fast to be able to observe the immediate power-up behavior without modification. The present technique proposes that the diagnostic circuitry should provide a mechanism whereby it automatically serves to prevent execution of program instructions (once any necessary program instructions or other operations have executed) upon power-up such that the diagnostic circuitry can be arranged and configured to monitor the power-up operations as desired.

It will be appreciated that in some embodiments program instruction execution can be prevented immediately power-up occurs with no program instructions being executed until released to do so by the diagnostic circuitry. Alternatively, in other embodiments, for example, overriding security considerations (e.g. a system which boots into a secure mode of operation and does not permit diagnosis during such a secure mode of operation) may necessitate at least some processing operations (e.g. changing to a non-secure mode of operation) to be performed prior to program instruction execution being prevented to allow the diagnostic system to be configured. Another example of processing operations which may be desired to be performed prior to program instruction execution being prevented is the restoration of state by the operating system following a power-down prior to diagnosis of the start up of an application program executing on that operating system layer. In this context, the operating system may be allowed to complete the restoration operation with the application program then being prevented from being executed by the diagnostic circuitry.

There are a number of ways in which the diagnostic circuitry can prevent execution of program instructions upon power-up. In one preferred technique the diagnostic circuitry generates a halt request which halts the data processing instruction execution. In another technique, the diagnostic circuitry may hold the data processing circuit in a reset state whereby program execution is prevented until release from that reset set is permitted.

Whilst it is possible that the diagnostic circuitry could be configured to always have the behavior of preventing program execution following power-up until a release is made, in preferred embodiments the diagnostic circuitry is configured by a diagnostic operation (which could be a bus signal, a diagnostic command, a diagnostic program instruction or other forms) such that it will give the above described behavior upon the next return to the powered up mode from the powered down mode.

In preferred embodiments the present technique may also be used to prevent program instruction execution following a warm reset during which the data processing circuit remains in the operational mode.

Viewed from another aspect the present invention provides a method of processing data, said method comprising the steps of:

5 executing program instructions with a data processing circuit to perform data processing operations during an operational mode, said data processing circuit being dynamically switchable between said operational mode and a powered down mode; and

 performing diagnostic operations upon said data processing circuit with a diagnostic circuit;

10 wherein when said data processing circuit returns to said operational mode from said powered down mode and any required processing operations have been completed, said diagnostic circuit prevents execution of further program instructions until released by said diagnostic circuit.

15 The above, and other objects, features and advantages of this invention will be apparent from the following detailed description of illustrative embodiments which is to be read in connection with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

20

 Figure 1 schematically illustrates an integrated circuit including a diagnostic mechanism and operable to support dynamic power switching;

 Figure 2 schematically illustrates a diagnostic interface circuit in more detail;

25

 Figure 3 is a signal timing diagram illustrating diagnostic forced entry using a halting mechanism upon power-up;

 Figure 4 is a timing diagram illustrating a diagnostic forced entry using an
30 extended reset signal; and

 Figure 5 is a diagram illustrating a diagnostic forced entry after register unlocking.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Figure 1 illustrates an integrated circuit 2 including a central processing unit 4 operable to perform data processing operations under control of program instructions. A memory system 6 supplies data operands and program instructions to the central processing unit 4. A dynamic power controller 8 is responsive to control signals generated by the central processing unit 4 to switch the central processing unit from an operational mode into a powered-down mode as demand requires.

A diagnostic interface 10 is coupled to an external diagnostic system 12 (e.g. a suitably programmed general purpose computer) to perform diagnostic operations upon the central processing unit 4. The diagnostic interface circuit 10 can respond to the external diagnostic system 12 to program diagnostic configuration data, such as breakpoints, watchpoints and diagnostic configuration parameters, to be stored within registers of the diagnostic interface circuit 10 itself or diagnostic registers embedded within the central processing unit 4.

Figure 2 schematically illustrates the diagnostic interface circuit 10 in more detail. A diagnostic controller 12 is responsive to diagnostic operation commands (e.g. signals upon a dedicated diagnostic peripheral bus, diagnostic program commands and the like to perform requested diagnostic operations). Diagnostic configuration registers 14 within the diagnostic interface circuit 10 store diagnostic configuration parameters. Registers 16 within the central processing unit 4 itself also store diagnostic configuration parameters, such as breakpoints and watchpoint data. These registers 16 are programmed via the diagnostic interface circuit 10 and lose their data when the central processing unit 4 is powered down by the dynamic power controller 8. Accordingly, if the immediate startup of the central processing unit 4 after power-up is to be subject to diagnostic operations, then the diagnostic interface circuit 10 and the register 16 within the central processing unit 4 need to be appropriately reprogrammed prior to the central processing unit 4 being allowed to start to execute the program instructions intended to follow the power-up.

Also shown within Figure 2 is a forced power-up diagnostic entry latch 16 which may be programmed by the diagnostic controller 12 to store a value indicating that when the next power-up occurs the diagnostic interface circuit 10 should prevent execution of program instructions by the processor core 4, other than those required (e.g. such as to switch out of a boot within a secure mode or to restore registers under operating system control for diagnosis of an application program) prior to diagnosis being performed upon the power-up code sequences and operation. The latch 16 may be considered as a flip-flop, a one bit register, a latch or other data storage element.

Figure 3 is a timing diagram illustrating a first example of how the system of Figures 1 and 2 may operate. In this example a halt upon power-up is forced. At point 18 a diagnostic operation command received by the diagnostic controller 12 serves to set the force power-up diagnostic entry latch 16 to trigger forced diagnostic entry upon the next power-up event. At point 20 the central processing unit 4 is performing normal processing operation and a power-down request is generated, such as a request from the operating system which has detected that the processing load is low and there is no requirement for the central processing unit 4 to continue to operate and consume power. At time 22 following a power-down request at time 20, normal operation ceases and power-down is triggered at point 24, which in turn triggers generation of a pending halt request at point 26 that is maintained throughout the power-down period. At time 28, the system emerges from power-down, for example due to a wake on interrupt arrangement, a watchdog or some other mechanism. Following the power-up at time 28, normal operation is resumed but the pending halt request serves to halt the normal operation as indicated at time 30. This allows the external diagnostic system 12 to reprogram the diagnostic configuration required to diagnose the power-up code sequences and operations. At time 32, the diagnostic programming has completed and the halt on power-up signal stored within the latch 16 can be cleared thereby releasing the halt request at time 34 which in turn allows the halting of program instruction execution to be cancelled at time 36 and normal operation with normal program execution to be resumed.

Figure 4 illustrates an alternative operating mechanism using an extended reset. In this example, the latch 16 being set at time 38 is used to indicate that the system

should be held in reset following a power-up event. Accordingly, when a power-down request at time 40 is subsequently followed by the power being removed from the central processing unit 4 at time 42, then this holding reset upon power-up requirement is still stored within the diagnostic interface circuit 10. When a subsequent power-up request occurs at time 44, the system forces a reset signal to be generated in the normal way such that the central processing unit 4 is held in reset whilst the power is actually restored at time 46. As indicated by the dotted line 48, the reset signal would normally be removed once the power signal has been safely restored and program instruction execution allowed to restart. However, the diagnostic interface circuit 10 under control of the hold in reset signal marker stored within the latch 16 serves to extend the reset period until this marker is cleared at time 50. This extended time allows the diagnostic mechanisms to be appropriately programmed to retrieve the desired diagnostic information from the power-up code sequences and operations. When the hold in reset flag is cleared this triggers the reset signal to be removed at time 52 and normal operation resumed at time 54.

Figure 5 schematically illustrates a further example embodiment. This example is similar to that of Figure 3 except that the signal stored within the latch 16 indicates that the central processing unit 4 should be halted once a register lock which has been placed by the operating system upon registers to be saved and restored across the power-down has been removed at time 56. This register locking and unlocking is performed under operating system control and helps to ensure that race conditions do not occur between the diagnostic mechanisms attempting to read or write these registers and the operating system saving and restoring these registers. Once the registers have been unlocked following the power-up, a halt request signal is generated at time 58 and this serves to halt the processing at time 60 despite the central processing unit being back in the normal operation state to allow the debug register to be configured. When the halt on unlock signal is cleared at time 62 under diagnostic operation command control, the halt request is then cleared at time 64 and the the halting of the central processing unit 4 is removed at time 66.

Although illustrative embodiments of the invention have been described in detail herein with reference to the accompanying drawings, it is to be understood that the

invention is not limited to those precise embodiments, and that various changes and modifications can be effected therein by one skilled in the art without departing from the scope and spirit of the invention as defined by the appended claims.